Nos. 2017-1118, 2017-1202

UNITED STATES COURT OF APPEALS
for the
FEDERAL CIRCUIT

ORACLE AMERICA, INC.,

Plaintiff-Appellant,

v.

GOOGLE INC.,

Defendant- Cross-Appellant.

Appeals from the United States District Court for the Northern District of California in Civil Action No. 3-10-CV-3561-WHA Judge William Alsup

BRIEF OF *AMICUS CURIAE* BSA | THE SOFTWARE ALLIANCE IN SUPPORT OF ORACLE AMERICA, INC.

Richard L. Rainey
Peter A. Swanson
COVINGTON & BURLING LLP
850 Tenth Street, NW
Washington, DC 20001-4956
202-662-6000

Counsel for Amicus Curiae BSA | The Software Alliance

CERTIFICATE OF INTEREST

Counsel for *Amicus Curiae* BSA | The Software Alliance certifies the following:

1. The full name of every *amicus* represented by me is:

   BSA | The Software Alliance.

2. The name of the real party in interest (if the party named in the caption is not the real party in interest) represented by me is:

   None.

3. All parent corporations and any publicly held companies that own 10 percent or more of the stock of the *amicus curiae* represented by me are:

   None.

4. The names of all law firms and the partners or associates that appeared for the *amicus* now represented by me in the district court or are expected to appear in this court are:

   BSA | The Software Alliance did not appear in the district court.

   Before this court, BSA | The Software Alliance is represented by Covington & Burling LLP and the following attorneys with that firm:

   Richard L. Rainey
   Peter A. Swanson

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

STATEMENT OF INTEREST

BSA | The Software Alliance ("BSA") is an association of the world's leading software and hardware technology companies.[1] On its members' behalf, BSA promotes policies that foster innovation, growth, and a competitive marketplace for commercial software and related technologies. BSA members hold a significant number of copyrights. And they rely on copyright protection to protect their critical assets. BSA members therefore have a strong stake in the proper functioning of the U.S. copyright system.

Given these facts, BSA is well situated to aid the Court in deciding a case that will help to define the contours of U.S. copyrightability and fair use law as applied to software. In this brief, BSA offers the Court a policy-based review of the legal issues at stake, emphasizing how they bear on the software industry.

[1] All parties have consented to BSA's filing this brief. Fed. R. App. P. 29(a)(2). No party's counsel authored this brief in whole or in part. No party, party's counsel, or any person other than *amicus* or its counsel contributed money intended to fund preparing or submitting this brief. *Id.* 29(c)(5).

SUMMARY OF ARGUMENT

As the Court considers this case, BSA[2] urges the Court to ground its decision in a holding that software is copyrightable, and to make sure that courts apply the fair use defense in software cases evenhandedly, along with other copyrighted works, based on a complete record.

In this case, the district court initially ruled that certain key features of Oracle's software—the structure, sequence, and organization ("SSO") of its Java programming language, and short lines of code used to declare methods—were not copyrightable. This Court reversed that determination.

Now the case is again before the Court, this time raising questions of whether the district court's application of the fair use defense was appropriate and whether intervening Ninth Circuit case law disturbs this Court's earlier ruling on copyrightability. BSA submits that the Court should reject any invitation to disturb its ruling based on recent Ninth Circuit case law. The software industry depends on copyright protection to drive innovation and economic growth. Having a reliable legal basis for protecting creativity at all levels of software development, and for promoting flexible licensing models and consumer choice, is critical to the health of the U.S. software industry.

[2] Microsoft and Salesforce.com, Inc. do not join in the filing of this amicus brief and their views are not reflected in the brief.

The Court should also ensure that courts applying fair use defenses to infringement in software cases do so correctly. Fair use may be important in various circumstances, but it should not be interpreted so broadly as to swallow the commercial value of an infringed underlying work by failing to fully and carefully weigh all four of the factors set out in 17 U.S.C. § 107. To protect the balance of this innovation equation, BSA urges the Court to follow well-established precedent: fair use is a case-by-case, fact-driven defense, so district courts should err on the side of allowing evidence to be presented to the jury. And merely adapting a copyrighted work, in this case a computer program, from a desktop to mobile environment presents a question of transformative use for which a jury must give due consideration to preexisting uses that are directly competitive, particularly where, as in this case, the copyrighted work was already used in smartphones before Android was released, the work was copied directly, and the copied code was used for the same purpose as Java.

ARGUMENT

I. Copyright Protection for Software Is Essential to Innovation and Economic Growth.

As a major component of the U.S. economy, the software industry depends on copyright protection to spur innovation and economic growth.

A. The Software Industry Is a Key Contributor to the U.S. Economy.

Software drives the modern economy in numerous ways. *First*, the software industry contributes more than a trillion dollars to the U.S. economy every year. BSA, *The $1 Trillion Economic Impact of Software*, at 3 (June 2016).[3] That number includes $475.3 billion in direct contributions and over $525 million in indirect and induced contributions. *Id*. The software industry makes these indirect contributions by supporting critical business functions, including "finance, human resources, operations and logistic, sale and market." Santanu Kumar Misra & Amitava Ray, "Integrated AHP-TOPSIS Model for Software Selection Under Multi-criteria Perspective," *Driving the Economy Through Innovation and Entrepreneurship* 879, 879 (2013).

Software creates efficiencies across nearly every aspect of the nation's economy, including the agricultural sector (where software helps farmers increase crop yields), the healthcare sector (where software drives diagnostic accuracy), and

[3] Available at http://softwareimpact.bsa.org/pdf/Economic_Impact_of_Software_Report.pdf (last accessed Feb. 12, 2017).

the public sector (where software helps governments deliver services, reduce traffic congestion, fight crime, and cut costs). BSA, *The $1 Trillion Economic Impact of Software*, at 7–9. Software also helps companies "collaborate more effectively internally and externally, scale operations faster, operate more efficiently, and innovate and experiment more strategically." The Boston Consulting Group, *The Great Software Transformation*, 11 (2013).[4] For example, software is now responsible for 80 percent of innovation in the automobile industry, just one of many sectors where products increasingly rely on software to improve efficiency, safety, and functionality. Peter Anden et al., *The Perils of Ignoring Software Development*, McKinsey Quarterly (Feb. 2015).[5]

Second, the software industry contributes significantly to national investment in research and development ("R&D"). It accounts for 17.2 percent of the nation's R&D spending: more than $50 billion dollars a year. BSA, *The $1 Trillion Economic Impact of Software*, at 4. At an annual growth rate of 13.2 percent, software is the fastest growing area of R&D spending in the entire economy. *Id.* U.S. software R&D spending now outstrips industrial R&D spending.

---

[4] Available at http://cpalemmens.com/mylibrary/Boston%20Consulting%20Group%20-%20The%20Great%20Software%20Transformation.pdf.

[5] Available at http://www.mckinsey.com/industries/high-tech/our-insights/the-perils-of-ignoring-software-development.

Third, the software industry provides nearly 10 million jobs for the American workforce. *Id.* at 3. That figure includes 2.5 million people directly employed by software companies, which pay salaries far above the national average. *Id.* Indeed, software developers earned an average $108,760 in 2014—more than twice the national average for non-software workers. *Id.* BSA has projected that these high-paying software jobs will grow at a rate of 3.1 percent through 2020. BSA, *Powering the Digital Economy: A Trade Agenda to Drive Growth*, 4 (2014).[6]

Also, software acts as a strong "employment multiplier" because every two jobs in software support an additional job in other industries. Robert J. Shapiro, *The U.S. Software Industry: An Engine for Economic Growth and Employment*, SIIA White Paper, 6-7 (2014).[7] Today, software indirectly supports an estimated 7.3 million jobs outside the software industry. BSA, *The $1 Trillion Economic Impact of Software*, at 3.

B. The Software Industry Relies on Copyright Protection.

Copyright protection is critical to the continued growth of the software industry itself, and the many sectors of the economy that depend on software to flourish. Because of the ease with which software may be copied, used

[6] Available at http://digitaltrade.bsa.org/pdfs/DTA_study_en.pdf.
[7] Available at https://www.siia.net/Admin/FileManagement.aspx/LinkClick.aspx?fileticket=yLPW0SrBfk4%3D&portalid=0.

and distributed, software authors need copyright protections for their works. Without such protections, any software product would be pilfered by users and competitors alike as soon as the first copy is released—and the software creator would be unable to recoup the cost of development.

Copyright also offers unique, flexible protection that is not available under other intellectual property regimes. Trade secret protection does not apply to publicly available source code (*e.g.*, as in some web-based technologies). Patents do not provide protection to software's expressive components—just its functional ones.

Copyright is the foundation of the software industry. For decades, software companies like BSA members have relied on the flexibility of copyright licenses to protect and distribute their products. Emery Simon, BSA, Testimony before the United States House of Representatives Committee on the Judiciary Subcommittee on Courts, Intellectual Property, and the Internet (June 2, 2014). Indeed, in 1969, when IBM first began to offer software separate from hardware, it used copyright licenses to do so. *Id*. Even as the model of software distribution has changed from physical sales to cloud-based licensing models, BSA members have continually relied on flexible—and reliable—copyright protection and licensing. Evolving distribution services like these, developed to meet the

demands of the modern software customer, can only succeed if copyright protection for the underlying software is reliably maintained at law.

II. Balanced Copyright Protection for Software Is Indispensable.

Although this appeal is focused on fair use, the Court should base its decision on earlier decisions regarding the copyrightability of software. And the Court should refuse any invitation, by Google or any amici, to undermine those decisions based on recent Ninth Circuit case law.

The Court should continue to follow the bedrock principle, stretching back over 40 years, that software is copyrightable if it is creative and original expression. In 1974, Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU) for the purpose of studying and compiling data on, *inter alia*, copyright protection for computer programs. *See* Pub. L. No. 93-573, § 201, 88 Stat. 1873 (1974); Final Report of the National Commission on New Technological Uses of Copyrighted Works, at 9 (1978) [hereinafter "CONTU Report"]. As this Court recognized in its earlier decision, "the thrust of the CONTU Report is that copyright is 'the most suitable mode of legal protection for computer software.'" *Oracle America, Inc. v. Google Inc.*, 750 F.3d 1339, 1380–81 (Fed. Cir. 2014) (quoting Peter S. Menell, *An Analysis of the Scope of Copyright Protection for Application Programs*, 41 Stan. L. Rev. 1045, 1072 (1989)); *see also* CONTU Report at 1 (recommending that copyright law be

amended "to make it explicit that computer programs, to the extent that they embody an author's original creation, are proper subject matter of copyright").

The CONTU Report formed much of the basis for the 1980 amendments to the Copyright Act that formally included computer software as copyrightable material. It emphasized that software—just like any other work of creative authorship—should be protectable if it is original. This approach embraced the rule of Section 102: any original work of authorship, fixed in any tangible medium of expression, is copyrightable. 17 U.S.C. § 102(a). Congress codified this approach in the 1980 amendments to the Copyright Act. It added computer programs as copyrightable subject matter, defining "computer programs" as "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101; Act of Dec. 12, 1980, Pub. L. No. 96-517, § 10, 94 Stat. 3015, 3028.

This history shows that computer programs are like any other copyrightable subject matter: if they are original, they are entitled to full copyright protection, subject to the standard exceptions to copyrightability. *Feist Publ'ns, Inc. v. Rural Tel. Serv. Co.*, 499 U.S. 340, 345 (1991) ("The *sine qua non* of copyright is originality."); *see also, e.g.*, 17 U.S.C. § 102(b); *Soc'y of Holy Transfiguration Monastery, Inc. v. Gregory*, 689 F.3d 29, 51–52 (1st Cir. 2012) (words and short phrases are generally excluded from copyright protection), *cert.*

denied (U.S. Nov. Feb. 19, 2013) (No. 12-7513); *Satava v. Lowry*, 323 F.3d 805, 812 & n.5 (9th Cir. 2003) (merger doctrine).

Courts have consistently applied these principles to find original computer programs to be copyrightable. Regardless of whether the computer program ("statements or instructions") is meant to do something functional, like solving a math problem, the expression of a program is protectable if its programmer wrote it in a creative way, provided that there were feasible alternative expressions available. *See, e.g., Apple Computer Inc. v. Franklin Computer Corp.*, 714 F.2d 1240, 1251 (3d Cir. 1983) (distinguishing between "the method which instructs the computer to perform its operating functions" and "the instructions themselves," determining the latter to be copyrightable).

Such functional uses do not preclude copyright protection for software under Section 102(b)—which provides that copyright does not extend to ideas, processes, and systems—any more than a user manual, however dry, takes a person instead of a computer through the steps for doing a task. *See, e.g.*, *M. Kramer Mfg. Co. v. Andrews*, 783 F.2d 421, 435 (4th Cir. 1986) (citing *Apple Computer* for this principle); *see also Williams Elecs. Inc. v. Artic Int'l Inc.*, 685 F.2d 870, 876–77 (3d Cir. 1982) (rejecting argument that "object code" was uncopyrightable simply because it could be understood only by a computer and explaining that 17 U.S.C. § 101 should not be interpreted "in a manner which would severely limit the

copyrightability of computer programs which Congress clearly intended to protect").

And of course, this protection extends not just to the software code itself, but also may apply to the structural elements of software—the SSO, for instance, as this Court knows. *Oracle*, 750 F.3d at 1368 (holding that because "the SSO is original and creative" and "the declaring code could have been written and organized in any number of ways and still have achieved the same functions," Section 102(b) does not bar the SSO or code comprising software packages from copyrightability just because they also perform functions); *see also, e.g.*, *Computer Assocs. Int'l v. Altai, Inc.*, 982 F.2d 693, 706–07 (2d Cir. 1992) (holding "structural components" subject to copyright protection if they reflect expressive elements).

These well-settled principles are consistent with a recent Ninth Circuit decision concerning copyrightability that was rendered after this Court's earlier decision in the present case, but that does not discuss this Court's decision at all. *Bikram's Yoga College of India, L.P. v. Evolation Yoga, LLC*, 803 F.3d 1032 (9th Cir. 2015). In *Bikram*, a yogi and his studio (the plaintiffs) maintained that a sequence of twenty-six yoga poses and two breathing exercises, as set forth in a book, was copyrightable. *Id.* at 1034. The plaintiffs had sued the defendants, rival yogis and their studio, for replicating the purportedly copyrighted sequence of

poses and exercises in their classes. *Id.* at 1036. (The plaintiffs did not allege the defendants had copied the book, just that they had replicated the poses and exercises in order.)

The Ninth Circuit ruled that the sequence was an unprotectable process despite any aesthetic or spiritual qualities it may have had. *Id.* at 1037–40. The court reasoned that because the sequence of poses and exercises was designed to warm, stretch, and strengthen the body, the sequence was a healing method and not protectable under Section 102(b). *Id.* at 1039 ("As the Supreme Court explained in *Baker*, 'Certain mixtures are found to be of great value in the healing art. If the discoverer writes and publishes a book on the subject (as regular physicians generally do), he gains no exclusive right to the manufacture and sale of the medicine; he gives that to the public.'") (quoting *Baker v. Selden*, 101 U.S. 99, 102–03 (1880)). This rule dates back more than a century to the Supreme Court's decision in *Baker*, which recognized that describing or instructing one how to perform some action is copyrightable, even if the underlying action is not:

> The description of the art in a book, though entitled to the benefit of copyright, lays no foundation for an exclusive claim to the art itself. The object of the one is explanation; the object of the other is use. The former may be secured by copyright. The latter can only be secured, if it can be secured at all, by letters-patent.

Baker, 101 U.S. at 105; *see also Bikram*, 803 F.3d at 1037–38 (basing its rule on *Baker*).

The distinction between the book in which the sequence was fixed, and the sequence itself, was critical. As *Bikram* notes, the book itself is still copyrightable even if the sequence itself is not: one could not reproduce it without the copyright holder's authorization. *See id.* at 1039–40. But anyone can perform the same or similar sequence without infringing the copyright in the book.

A computer program is analogous to the book from *Bikram*. A computer program instructs a computer to perform a function: sorting numbers by size, for instance. That function itself could never be copyrightable, any more than the accounting principles described in the book in *Baker* or the yoga sequence from *Bikram*. But the computer program could be protectable if it is original, just like the books in *Bikram* and *Baker*. *See Oracle*, 750 F.3d at 1367 ("[W]e conclude that a set of commands to instruct a computer to carry out desired operations may contain expression that is eligible for copyright protection."). *Bikram* does not make a new rule for software. It simply maintains the division between creative ways of describing functional processes—basically what software does—and the processes themselves.

The same principles of copyrightability apply equally to the literal elements of a computer program, such as declaring code and method code, and to its non-literal elements like the SSO. Just as the non-literal elements of a book may be copyrightable, the SSO and other non-literal elements of software may be

copyrightable if they are minimally original and creative. *Oracle*, 750 F.3d at 1354–56. As this Court previously explained, both the SSO and the declaring code reflected a creative and expressive way of describing the functional aspects of the Java programming language. *Id.* at 1367–68.

This is unlike *Bikram*, where the yogi was not suing the defendants for copying any literal or non-literal expression in his book, but for copying the underlying functionality itself—*i.e.*, his sequence of poses and exercises. By copying Oracle's API packages, Google copied expression in Oracle's program, not merely the function of the program. That is the critical difference. This Court previously held that Oracle's API packages are not just some abstract structure or process (*e.g.*, the steps of a math problem, or the sequence from *Bikram* apart from the book) but an actual work, parts of which Google copied even if it rewrote other parts.

For these reasons, the Court should not entertain any invitation to disturb or upend its 2014 opinion in this case on the basis of *Bikram*, which neither cited nor discussed this Court's prior opinion.

III. Fair Use Must Be Applied Correctly Upon a Full Record.

Just as copyrightability must be applied correctly, so too must fair use be properly and carefully considered. BSA offers two policy considerations for the Court's analysis of fair use in this case.

First, courts recognize uniformly that evaluating the fair use defense is case-by-case and fact-driven. It is not to be simplified with bright-line rules. *Harper & Row Publ'rs v. Nation Enters.*, 471 U.S. 539, 560 (1985). As the Supreme Court long ago recognized, fair use requires courts to apply an "equitable rule of reason" to fair use analysis. *Sony Corp. of Am. v. Universal City Studios, Inc.*, 464 U.S. 417, 448 (1984). This requires carefully weighing the record evidence and the fair use factors together, in light of the purposes of copyright. *E.g.*, *Wright v. Warner Books, Inc.*, 953 F.2d 731, 740 (2d Cir. 1991) ("The fair use test remains a totality inquiry, tailored to the particular facts of each case.").

Given the fact-dependent nature of fair use, the finder of fact—whether judge or jury—must have all relevant evidence presented. As the Supreme Court noted in considering the purpose or use of a work, which is one factor identified in the statute, *see* 17 U.S.C. § 107, courts should "preserve the breadth of their traditionally ample view of the universe of relevant evidence." *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 584 (1994) (citing *Harper & Row*, 471 U.S. at 561). The same is true for the fair use analysis more generally. *See id.* at 584–92 (urging broad evidentiary considerations throughout fair use analysis).

Indeed, the statute itself makes clear that the inquiry is open-ended: Section 107 provides that the four factors are not exclusive, but serve as guideposts

for the analysis. *See* 17 U.S.C. § 107 (stating that the determination must "include" the four factors); *see also, e.g.*, *Harper & Row*, 471 U.S. at 549 ("Section 107 requires a case-by-case determination whether a particular use is fair, and the statute notes four nonexclusive factors to be considered."). Given how broadly fair use considerations can sweep, having a full evidentiary record is critical to ensuring that the fair use defense to infringement serves its statutory purpose.

Although the district court has discretion in determining what evidence is presented, the structure of the fair use statute should favor admissibility. In this case, the Court should review the full record to ensure that the district court admitted evidence that is probative of any of the four factors set out in the statute, or any other relevant considerations given these factors are not exhaustive. As discussed in Oracle's opening brief, the district court excluded evidence of Google's uses of Oracle's APIs for devices other than smartphones and tablets. Oracle Br. 55. As a result, not all of the probative evidence was made available to the jury. Regardless of the ultimate outcome, admitting as much evidence as possible would better preserve "the breadth of [the courts'] traditionally ample view of the universe of relevant evidence" for fair use. *Campbell*, 510 U.S. at 584 (citing *Harper & Row*, 471 U.S. at 561). By applying these principles, courts can ensure that fair use only applies when it is factually

appropriate: not every case is the same, nor is every computer program, and they should not be treated the same.

Second, in this context, the Court should carefully consider whether there was adequate evidence presented from which a reasonable jury could have found a fair use in this case. In particular, this Court should review the evidence to determine whether there were adequate facts presented from which a reasonable jury could find a transformative use of the copyright work under the first factor. The Supreme Court has held that a use is not transformative when "the new work merely supersedes the objects of the original creation." *Campbell*, 510 U.S. at 579 (internal quotation marks and alterations omitted). This Court previously held that such superseding uses exist "[w]here the use is for the same intrinsic purpose as the copyright holder's." *Oracle*, 750 F.3d at 1375 (internal quotations omitted). As Oracle noted in its opening brief, the district court, Google itself, and Google's expert witness agreed that the copied APIs "serve the same function in both works." Oracle Br. 31. The district court nevertheless held that the jury could have found Google's use of the copyrighted works to be transformative for having moved from a desktop to a mobile operating system. But the jury would have had to ignore the undisputed facts that the copyrighted work was already used in smartphones before Android was released and that the work was both copied

directly and used for the same purpose and with the same expression (the declaring code) as Java.

Making a program developed for a particular architecture also compatible with other systems generally requires copying and modifying the software's code. Sometimes these modifications are minimal. Sometimes they are extensive. But they do not change the fundamental purpose for that software. *See Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 819 (9th Cir. 2003) ("Courts have been reluctant to find fair use when an original work is merely retransmitted in a different medium."); *see also Infinity Broad. Corp. v. Kirkwood*, 150 F.3d 104, 108 n.2 (2d Cir. 1998) (holding that a "change of format" does not make a use transformative); *Am. Geophysical Union v. Texaco Inc.*, 60 F.3d 913, 923 (2d Cir. 1994) (noting that an "untransformed copy is likely to be used simply for the same intrinsic purpose as the original, thereby providing limited justification for a finding of fair use").

The district court in this case held that Google's use of Oracle's code for "one component in a full-stack platform for highly advanced smartphones" could be a transformative use of Oracle's code because that code was designed for desktop and laptop computers. Appx. 46. But as this Court noted in 2014, the record shows that Java had been licensed for use on mobile phones years before the launch of Android or the commencement of this dispute. *Oracle*, 750 F.3d at

1350 ("The testimony at trial also revealed that Sun was licensing a derivative version of the Java platform for use on mobile devices: the Java Micro Edition ('Java ME'). Oracle licensed Java ME for use on feature phones and smartphones."). So in this case, the jury in the second trial would have had to fail to properly consider the undisputed fact that Java was already used in smartphones before Android was released.

CONCLUSION

For the foregoing reasons, *amicus curiae* BSA urges the Court to affirm the applicability of fundamental principles of copyrightability and fair use law in ruling on this appeal and to confirm that such principles apply equally to all works of authorship including computer programs. The Court should ensure that software companies like BSA members can continue to rely on copyright protection for their valuable original works.

Respectfully submitted,

/s/ Richard L. Rainey
Richard L. Rainey
Peter A. Swanson
COVINGTON & BURLING LLP
850 Tenth Street, NW
Washington, DC 20001-4956
202-662-6000

Counsel for Amicus Curiae
BSA | The Software Alliance

CERTIFICATE OF SERVICE

I certify that I served a copy on counsel of record on February 17, 2017, by Electronic Means by CM/ECF.

/s/ Richard L. Rainey
Richard L. Rainey
COVINGTON & BURLING LLP
850 Tenth Street, NW
Washington, DC 20001-4956
202-662-6000
rrainey@cov.com

Counsel for Amicus Curiae
BSA | The Software Alliance

CERTIFICATE OF COMPLIANCE
WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE REQUIREMENTS

I certify that the foregoing brief complies with the type-volume limitation of Federal Rules of Appellate Procedure 29(a)(5) and 32(a) and Federal Circuit Rule 28.1. This brief contains 3,979 words, excluding the parts of the brief exempted by Federal Rule of Appellate Procedure 32(f).

This brief complies with the typeface requirements of Federal Rule of Appellate Procedure 32(a)(5) or Federal Circuit Rule 28.1 and the type style requirements of Federal Rule of Appellate Procedure 32(a)(6). This brief has been prepared in a proportionally spaced typeface using Microsoft Word in 14 point Times New Roman font.

/s/ Richard L. Rainey
Richard L. Rainey
COVINGTON & BURLING LLP
850 Tenth Street, NW
Washington, DC 20001-4956
202-662-6000
rrainey@cov.com

Counsel for Amicus Curiae
BSA | The Software Alliance